

How to change PostgreSQL's data directory on Linux

Mar 25, 2021

There comes a time when you have to restore a relatively large database locally. Most likely, you've partitioned your disk, and your `root` partition got the thick end of it, 50 GB if you were generous. Let's assume that that's not nearly enough for the database you're about to restore. At the same time, your `/home` partition got the rest of the disk space you had available.

You can always resize these two partitions, but then you would have back up your `/home` directory, unmount it or find a Live USB and tamper with them. If you don't have the time, or the desire, or even a backup disc, you can always change the location where `postgresql` stores its data.

The following instructions are a love letter to all those lost souls who find themselves in this situation and forget to check the status of SELinux, as well as to my future self who'll most likely have to do it again. Considering this is mostly a dump of my bash history that has everything and anything, I hope this exact procedure works for you. If not, feel free to contact me and we'll update it together.

Procedure

A fresh install is the easiest to change, but let's assume you have some databases locally you don't want to lose, just move them and restore the large database next to them. The steps are more or less the same anyway.

`data_directory` and `config_file`

Before you start anything, locate the `postgresql`'s configuration file and its data directory:

```
$ sudo su - postgres
[postgres@host ~]$ psql
Password for user postgres:
psql (12.6)
Type "help" for help.

postgres=# SHOW config_file;
```

```
config_file
```

```
-----  
/var/lib/pgsql/data/postgresql.conf  
(1 row)
```

```
postgres=# SHOW data_directory;  
data_directory
```

```
-----  
/var/lib/pgsql/data  
(1 row)
```

Stop the `systemd` service

Stop the `postgresql` `systemd` service:

```
systemctl stop postgresql.service
```

New location

Create a directory where you have enough disk space available (in this case, it's the `/home` directory), grant the `postgres` user ownership and permissions over it and copy the original data directory to the new location (the key is to [preserve the same ownership and permissions structure](#)):

```
mkdir /home/pgdata  
chown postgres:postgres /home/pgdata  
chmod 700 /home/pgdata  
rsync -av /var/lib/pgsql/data/ /home/pgdata/data
```

`postgresql` configuration

Open the `postgresql.conf` file in the new location and update the `data_directory` variable, setting it to the new location where your data was moved:

```
vim /home/pgdata/data/postgresql.conf
```

```
#-----  
# FILE LOCATIONS  
#-----  
  
# The default values of these variables are driven from the -D command-  
# option or PGDATA environment variable, represented here as ConfigDir.
```

```
data_directory = '/home/pgdata/data' # use data in another directory
# (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf' # host-based authentication file
# (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
```

systemd configuration

Do the same thing with the `postgresql.service`'s `systemd` configuration file:

```
vim /lib/systemd/system/postgresql.service
```

```
# ...
Environment=PGDATA=/home/pgdata/data
# ...
```

Once you're done editing the `systemd` configuration, reload it and start the service:

```
systemctl daemon-reload
systemctl start postgresql.service
systemctl status postgresql.service
```

SELinux

If you're receiving some vague `Permission denied` errors, [check whether or not you have SELinux enabled](#):

```
cat /sys/fs/selinux/enforce
1
```

If the result is `1`, then SELinux is in `enforcing` mode. To temporarily [set it to permissive mode \(0\)](#), run:

```
setenforce 0
```

You can try starting the `postgresql.service` again. If the process has started successfully, stop it, and tell SELinux to [apply the same context to the new location](#). Then you can return SELinux to `enforcing` mode

```
semanage fcontext --add --equal /var/lib/pgsql /home/pgdata
restorecon -rv /home/pgdata
setenforce 1
```

Then you should be able to start the `postgresql.service` without any errors

```
systemctl start postgresql.service
systemctl status postgresql.service
```

Confirmation

To confirm the new location and configuration is used, rerun the first step:

```
→ sudo su - postgres
[postgres@lenovo ~]$ psql
Password for user postgres:
psql (12.6)
Type "help" for help.

postgres=# SHOW data_directory;
 data_directory
-----
 /home/pgdata/data
(1 row)

postgres=# SHOW config_file;
 config_file
-----
 /home/pgdata/data/postgresql.conf
(1 row)
```

fitodic's blog

fitodic's blog

 [fitodic](#)

 [fitodic](#)

 [fitodic](#)

Copyright (c) 2019-present Filip Todić.
Content licensed under the Creative
Commons Attribution-NonCommercial-
ShareAlike licence.